Mohammad H. Tabatabaei

Smart Contracts

Smart contracts are user-defined programs that specify rules governing transactions and that are enforced by a network of peers. Smart contract programming is similar to traditional programming; however, it introduces some new security challenges that can harm the system more than before. If you inject money into a buggy smart contract, you will probably lose it. Contracts must be written to ensure fairness even when counterparties may attempt to cheat in arbitrary ways that maximize their economic gains.

A contract storage file is stored on the public blockchain and allocates space for data such as public keys of senders or receivers, or the deadline of the swap contract. A smart contract program is executed by a network of miners who reach consensus on the outcome of the execution and update the contract's state on the blockchain accordingly. The contract's code can be executed whenever it receives a message from a user or another contract. Then, it may read from or write to its storage file. Thus, transactions act like function calls in traditional programming languages. A contract processes a message that it receives, then it can pass a return value back to the sender. It should be considered that Ethereum uses the concept of gas to discourage over consumption of resources.

Smart contract programming needs more precautions due to its economic inherent. There is more chance for occurring pitfalls in smart contract programming. In a decentralized cryptocurrency, multiple parties may be sending inputs to the contract simultaneously. In this case, it is up to the miner who mines this block to decide how to order these transactions. Therefore, because of the wrong programming, the possibility of inaccessibility to the sent money or money leakage by the contract exists. In addition, if the counterparties of the contract send their inputs in cleartext, they are potentially targeted by malicious parties. Finally, some bugs are derived inherently from the implementation of the cryptocurrency. For example, Ethereum limits the resulting call-stack to a fixed size of 1024. Therefore, if the call-stack depth is already at this limit when the send a message to another contract instruction is reached, then that instruction will be skipped.

At the end, a properly written smart contract should describe all possible outcomes of the contract. Furthermore, a smart contract is deterministic and the same input will always produce the same output. If one writes a non-deterministic contract, when it is triggered it will execute on every node on the network and may return different random results, thus preventing the network from reaching consensus on its execution result. To put it in a nutshell, smart contracts operate as autonomous actors in the blockchain, whose behaviour is completely predictable.

- Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab
- Blockchains and Smart Contracts for the Internet of Things
- Making smart contracts smarter